UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/719,673	11/21/2003	Khosro Khakzadi	03-1862/L13.12-0251	1314

7590   12/28/2007

Leo Peters
LSI Logic Corporation
MS D-106
1621 Barber Lane
Milpitas, CA 95035

EXAMINER
WIENER, ERIC A

ART UNIT	PAPER NUMBER
2179	

MAIL DATE	DELIVERY MODE
12/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _08 October 2007_.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,5-16 and 19-24_ is/are pending in the application.

   4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1,5-16 and 19-24_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _11/21/2003_ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All   b)☐ Some *   c)☐ None of:

    1.☐ Certified copies of the priority documents have been received.

    2.☐ Certified copies of the priority documents have been received in Application No. _____.

    3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1.   A request for continued examination under 37 CFR 1.114, including the fee set forth in

37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible

for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been

timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR

1.114. Applicant's submission filed on 10/8/2007 has been entered.

2.   Claims 1, 5 – 16, and 19 – 24 are pending. Claims 2 – 4 have been cancelled. Claims 1, 8,

19, and 24 are the independent claims. Claims 1, 7, 8, 23, and 24 are the amended claims. Claims

1, 5 – 16, and 19 – 24 are rejected by the examiner.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4.   Claims 1 – 4, 7 – 13, and 19 – 20 are rejected under 35 U.S.C. 102(b) as being anticipated

by Nahaboo et al. (US 5,974,253).

As per claim 1, Nahaboo discloses *a command processor stored on a computer readable*

memory for use with a computer system (column 6, lines 16 – 18) *comprising a graphical user*

interface for providing a graphical interface to the computer system (Fig. 2) *and a command*

interpreter (column 6, line 50), *which:*

- *loads one or more configuration commands into the command processor from*

  at least one of:

  o *a user specified command configuration script comprising the one or more*

    configuration commands (column 6, lines 50 – 59 and column 13, lines 32 –

    35), wherein command configuration scripts are part of interface description

    files, which are user specified because a user may specify an interface

    description file to load, *or from*

  o *a command line in which the one or more configuration commands are*

    entered by the user (column 9, lines 53 – 60, column 10, lines 36 – 44, and

    Fig. 4B), wherein the command line area to enter configuration commands is

    area 315 of Fig. 4B, *and*

- *interprets the configuration commands and modifies the graphical user*

  interface at run time of the graphical user interface according to the interpreted

  configuration commands (column 6, lines 50 – 64), wherein the fact that the

  interpreter interprets **all** events that arise and that the system reacts in a purely

  dynamic manner means that the interpreter will interpret both saved user

  specified configuration scripts and configuration commands entered by a user

  from a command line in order to dynamically modify the interface, *including:*

  o *building graphical objects according to the interpreted configuration*

    commands (column 7, line 55 – column 8, line 10),

o *assigning functionality to the built graphical objects according to the interpreted configuration commands* (column 9, lines 39 – 42), *and*

o *displaying a user-interactive window containing the graphical objects according to the interpreted configuration commands* (column 10, line 45 - column 11, line 2, column 15, lines 20 – 21, and Fig. 5), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface.

As per claim 7, and taking into account the rejection of claim 1, Nahaboo further discloses *a graphics engine tool for drawing contents of a database into the graphical user interface based on the user configuration commands* (column 3, lines 52 – 61).

As per claim 8, Nahaboo discloses *a method of providing a fully customizable graphical user interface* (column 6, lines 11 – 23) *comprising:*

- *upon execution of a command processor, loading a top level Tool Command Language (TCL) command into a namespace* (column 3, lines 52 – 54 and column 6, lines 50 – 56)

- *loading one or more TCL commands into the command processor from at least one of:*

    o *a user specified TCL command configuration script comprising the one or more TCL configuration commands* (column 6, lines 50 – 59 and column 13, lines 32 – 35), wherein command configuration scripts are part of interface

description files, which are user specified because a user may specify an interface description file to load, *or from*

o *a command line in which the one or more TCL configuration commands are entered by the user* (column 9, lines 53 – 60, column 10, lines 36 – 44, and Fig. 4B), wherein the command line area to enter configuration commands is area 315 of Fig. 4B, *and*

- *building graphical objects* (column 7, line 55 – column 8, line 10) *according to the TCL configuration commands* (column 6, lines 53 – 56)

- *assigning functionality to the built graphical objects* (column 9, lines 39 – 42) *according to the TCL configuration commands* (column 6, lines 53 – 56)

- *displaying a user-interactive window containing the graphical objects according to the TCL configuration commands* (column 3, lines 34 – 36), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface

The examiner has interpreted the fact that Nahaboo discloses using an interpreted language (Abstract, lines 1 – 2) sufficiently discloses the use of the interpreted language Tool Command Language.

As per claim 9, and taking into account the rejection of claim 8, Nahaboo further discloses *performing functions based on user interactions with the graphical objects according to their assigned functionality* (column 10, lines 45 – 54).

As per claim 10, and taking into account the rejection of claim 8, Nahaboo further discloses that *the graphical objects are selected from a group consisting of windows, window panes* (column 4, lines 39 – 41), *buttons, and menus* (column 6, lines 4 – 5).

As per claim 11, Nahaboo substantially discloses the method of claim 8. In addition, Nahaboo further discloses *creating the TCL command configuration script and assigning the TCL command configuration script to one of the graphical objects* (column 11, lines 55 – 58). Nahaboo does not explicitly disclose the TCL script corresponds to a circuit design function. However, Nahaboo does disclose that "the purpose of this invention is to define an extremely flexible interface development tool that can be used regardless of the application" (column 1, lines 29 – 31). Therefore, Nahaboo's script could correspond to a circuit design function if the user defines it in such a way.

As per claim 12, and taking into account the rejection of claim 11, Nahaboo further discloses that *one of the graphical objects is a button* (column 6, line 4).

As per claim 13, and taking into account the rejection of claim 11, Nahaboo further discloses that *one of the graphical objects is an item within a pull-down menu* (column 6, line 5).

As per claim 19, Nahaboo discloses *a method of providing a graphical user interface, comprising:*

- *loading a top level Tool Command Language (TCL) command into a namespace upon execution of a command processor* (column 3, lines 52 – 54 and column 6, lines 50 – 56)

- *providing a command interpreter for interpreting one or more configuration commands from a user* (column 6, liens 50 – 52)

- *loading the one or more configuration commands into the command processor from*

  at least one of:

  o *a user specified command configuration script comprising the one or more*

  configuration commands (column 6, lines 50 – 59 and column 13, lines 32 –

  35), wherein command configuration scripts are part of interface description

  files, which are user specified because a user may specify an interface

  description file to load, *or from*

  o *a command line in which the one or more configuration commands are*

  entered by the user (column 9, lines 53 – 60, column 10, lines 36 – 44, and

  Fig. 4B), wherein the command line area to enter configuration commands is

  area 315 of Fig. 4B, *and*

- *assembling a graphical user interface having no hard coded objects based on the*

  interpreted configuration commands from the user (column 6, lines 53 – 59)

- *wherein all objects within the graphical user interface are user defined through the*

  one or more configuration commands (column 6, lines 53 – 59), and wherein the fact

  that the user can enter the 'editing' mode without entering the 'execution' mode

  means that said user can define all objects of the graphical user interface before

  execution, thus defining all objects of a graphical user interface having no hard coded

  objects.

The examiner has interpreted the fact that Nahaboo discloses using an interpreted

language (Abstract, lines 1 – 2) sufficiently discloses the use of the interpreted language Tool

Command Language.

As per claim 20, and taking into account the rejection of claim 19, Nahaboo further

discloses *changing the graphical user interface based on changing configuration commands*

from the user (column 6, lines 62 – 63) *and displaying a changed graphical user interface*

during operation based on the changing configuration commands (column 6, lines 56 – 59).

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

6.   This application currently names joint inventors. In considering patentability of the

claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various

claims was commonly owned at the time any inventions covered therein were made absent any

evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out

the inventor and invention dates of each claim that was not commonly owned at the time a later

invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c)

and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7.   Claims 5 – 6, 14 – 16, and 21 – 24 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Nahaboo et al. (US 5,974,253) and Dangelo et al. (US 5,493,508).

As per claim 5, Nahaboo substantially discloses the command processor of claim 1.

However, Nahaboo does not explicitly disclose a suite of integrated circuit tools, each design

tool of the suite having a functionality corresponding to one or more steps in a design flow process of an integrated circuit.

On the other hand, in an analogous art, Dangelo discloses *a suite of integrated circuit tools, each design tool of the suite having a functionality corresponding to one or more steps in a design flow process of an integrated circuit* (column 1, lines 7 – 9, 38 – 41).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teaching of Dangelo into the command processor of Nahaboo to develop a command processor comprising a suite of integrated circuit design tools with a modifiable graphical user interface. The modification would have been obvious, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65).

As per claim 6, Nahaboo and Dangelo substantially disclose the command processor of claim 5. In addition, Nahaboo further discloses that *the command processor loads each design tool into the graphical user interface based on the user configuration commands* (column 6, lines 50 – 52).

As per claim 14, Nahaboo substantially discloses the method of claim 8. However, Nahaboo does not explicitly disclose changing a look and feel of the graphical user interface during a design process.

On the other hand, in an analogous art, Dangelo discloses *changing a look and feel of the graphical user interface during a circuit design process* (column 2, lines 62 – 65).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teaching of Dangelo into the method of Nahaboo to develop a method of providing a fully customizable graphical user interface for customizing during a circuit design process. The modification would have been obvious, because the nature of utilizing a graphical user interface in a circuit design process would allow for the modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65).

As per claim 15, Nahaboo and Dangelo substantially disclose the method of claim 14. In addition, Nahaboo further discloses *creating new graphical objects, previously undefined by the command processor* (column 7, line 55 – column 8, line 10) *using the TCL configuration commands* (column 6, lines 53 – 56) *and assigning functionality to the new graphical objects* (column 9, lines 39 – 42).

As per claim 16, Nahaboo and Dangelo substantially disclose the method of claim 14. In addition, Nahaboo further discloses:

- *loading a new top level TCL command into the namespace which corresponds to one or more new TCL configuration commands* (column 3, lines 52 – 54, column 6, lines 50 – 56, and column 9, lines 53 – 60)

- *building graphical objects* (column 7, line 55 – column 8, line 10 and column 10, lines 36 – 44) *according to new TCL configuration commands* (column 6, lines 53 – 56)

- *assigning functionality to the built graphical objects* (column 9, lines 39 – 42 and column 10, lines 36 – 44) *according to the new TCL configuration commands* (column 6, lines 53 – 56)

- *displaying the user-interactive window containing the graphical objects according to the new TCL configuration commands* (column 3, lines 34 – 36), wherein the disclosed example is not intended to limit the invention in any way, and is only an example of one of many different types of GUI's that may be created from objects built through configuration commands as made possible through Nahaboo's disclosed invention for developing a user interface

The examiner has interpreted the fact that the interface can load the file of commands to reconstruct the interface in a purely dynamic manner to be sufficiently equivalent to being able to change a look and feel according to new commands and objects.

As per claim 21, Nahaboo substantially discloses the method of claim 19. Nahaboo does not explicitly disclose interfacing with a suite of integrated circuit design tools for producing an integrated circuit layout and associated netlist.

However, in an analogous art, Dangelo discloses *interfacing with a suite of integrated circuit design tools for producing an integrated circuit layout and associated netlist* (column 2, line 55 – column 3, line 13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teaching of Dangelo into the method of Nahaboo to develop a method of providing a graphical user interface having no hard coded objects for interfacing with a suite of integrated circuit design tools. The modification would have been obvious, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface to produce a design (Dangelo, column 2, lines 62 – 65). If the user has control over defining some aspects of an interface, it

would be obvious that the user would have the ability to completely define all aspects of an interface for circuit design.

As per claim 22, Nahaboo and Dangelo substantially disclose the method of claim 21. In addition, Nahaboo further discloses *loading a design tool from the suite of design tools into the graphical user interface based on a user command* (column 8, lines 7 – 51).

As per claim 23, Nahaboo and Dangelo substantially disclose the method of claim 22. In addition, Nahaboo further discloses that *the one or more user configuration commands are assigned to one or more graphical objects* (column 9, lines 39 – 42).

As per claim 24, Dangelo discloses *an integrated circuit software design suite* (column 5, lines 21 – 26) *comprising:*

- *a command processor having a graphical user interface* (column 8, lines 47 – 67) *and a command interpreter* (column 5, lines 25 – 28) *for interpreting user commands* wherein the fact that the system processes user selections for graphically interfacing with elements is equivalent to having a processor for processing and interpreting commands

- *one or more design tools corresponding to processes within an integrated circuit design process* (column 8, lines 47 – 50)

- *wherein the one or more design tools operate under control of the command processor and within the graphical user interface* (column 8, lines 47 – 67), *further wherein the configuration commands build graphical objects for the graphical user interface and assign functionality to the built graphical objects* (column 8, lines 47 – 67).

Dangelo does not explicitly disclose that the graphical user interface is specified entirely by a user at runtime.

However, in an analogous art, Nahaboo discloses *a graphical user interface specified entirely by a user through one or more configuration commands loaded into the command processor at run time of the command processor and interpreted by the command interpreter* (column 6, lines 56 – 64), *wherein the configuration commands build graphical objects for the graphical user interface* (column 7, line 55 – column 8, line 10) *and assign functionality to the built graphical objects* (column 9, lines 39 – 42).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the teaching of Dangelo into the software design suite of Nahaboo to develop an integrated circuit software design suite comprising a graphical user interface specified entirely by a user at run time. The modification would have been obvious, because the nature of utilizing a graphical user interface in a circuit design process would allow for the specification and modification of the graphical user interface at runtime (Dangelo, column 2, lines 62 – 65). If the user were able to specify parts of the interface at runtime, it would be obvious that said user would have the ability to specify the entire interface at runtime.

Response to Arguments

8.   Applicant's arguments filed on 10/8/2007 have been fully considered but they are not persuasive.

9.   The applicant has argued that Nahaboo et al. do not enable modifying a GUI at run time

of that interface.

In response to this argument, please refer to the rejection of claim 1 *supra* and also to

column 6, lines 50 – 59 of Nahaboo et al., wherein it states that the interpreter "interprets all

events that arise from the MOTIF toolkit" and that "the system will react in a purely dynamic

manner." This has been interpreted as sufficiently equivalent to modifying at run time.

10.   The applicant has argued that Nahaboo et al. do not disclose performing such

modifications through configuration commands that are interpreted by a command processor.

In response to this argument, please refer to the rejection of claim 1 *supra* and also to

column 6, lines 50 – 64 of Nahaboo et al., wherein it states that the interpreter "interprets all

events that arise from the MOTIF toolkit" and that "in the editing mode, the application interface

can easily be modified." This has been interpreted as sufficiently equivalent to performing

modifications through configuration commands that are interpreted by a command processor.

11.   The applicant has argued that Nahaboo et al. do not disclose loading "configuration

<u>commands</u> into the command processor, at least one of a user specified command configuration

script comprising the one or more configuration commands," as recited by claim 1.

In response to this argument, please refer to the rejection of claim 1 *supra* and also to

column 6, lines 50 – 67 of Nahaboo et al., wherein it has been interpreted that the interface

description file that is loaded sufficiently contains configuration commands that correspond to

commands pertaining to the operations of the EGERIA editor, used for configuring.

12.   The applicant has argued that the "script" mentioned by Nahaboo et al. does not

correspond to configuration commands that modify the graphical user interface at run time of the

interface.

In response to this argument, please refer to the rejection of claim 1 *supra* and also to

column 6, lines 50 – 67 of Nahaboo et al., wherein it has been interpreted that a script is

sufficiently equivalent to and/or a part of an interface description file, wherein said interface

description file modifies the GUI at runtime as disclosed in section 9., *supra*.

13.   The applicant has argued that Nahaboo et al. do not disclose loading "configuration

commands into the command processor... from a command line... entered by the user," as

recited by claim 1, wherein those configuration commands modify the GUI at run time of the

interface.

In response to this argument, please refer to the rejection of claim 1 *supra* and also to

column 9, lines 53 – 60; column 10, lines 36 – 44; and Fig. 4B, wherein the command line area

to enter configuration commands is area 315 of Fig. 4B, further wherein those configuration

commands modify the GUI at runtime as disclosed in section 9., *supra*.

14.   The applicant has argued that Nahaboo et al. do not disclose configuration commands

(loaded at run time) modify the interface by: building graphical objects according to the

interpreted configuration commands, assigning functionality to the built graphical objects

according to the interpreted configuration commands, and displaying a user-interactive window

containing the graphical objects according to the interpreted configuration commands.

In response to this argument, please refer to the new rejection of claim 1 *supra* and also

to column 7, line 55 – column 8, line 10; column 9, lines 39 – 42; and column 10, lines 36 – 44,

wherein the creation of objects has been interpreted as sufficiently equivalent to building objects,

and further wherein the commands pertaining to said creation, which assign functionality, may

be interpreted configuration commands, such as those entered by a user.

In addition, please refer to column 10, line 45 – column 11, line 2; column 15, lines 20 –

21; and Fig. 5, wherein an example of displaying a user-interactive window containing the

graphical objects according to the interpreted configuration commands is disclosed, further

wherein the disclosed example is not intended to limit the invention in any way, and is only an

example of one of many different types of GUI's that may be created from objects built through

configuration commands as made possible through Nahaboo's disclosed invention for

developing a user interface.

15. The applicant has argued that Nahaboo et al. do not allow for objects to be built and

functionality to be assigned in the manner recited in claim 8.

In response to this argument, please refer to the rejection of claim 8 *supra*. In addition,

the disclosed sections of Nahaboo et al. pertaining to claim 8, sufficiently teach building and

assigning functionality to objects, wherein the use of an object toolbox, such as the disclosed

X/MOTIF graphical object toolbox, has been interpreted as sufficiently equivalent to the use of a

Tool Command Language, wherein both are languages that may be interchangeably used for

creating, modifying, and displaying an interface, and in either case will achieve the same desired result.

16.   The applicant has argued that the examiner seems to ignore the context of claim 19 in which the objects are defined through <u>configuration commands</u> that are loaded into a command interpreted through a script of such commands or a command line.

In response to this argument, please refer to the rejection of claim 19 *supra* and also to column 9, lines 39 – 42; and column 10, lines 36 – 44, wherein the defining of objects has been interpreted as sufficiently equivalent to building objects, and further wherein the commands pertaining to said defining, which assign functionality, may be interpreted configuration commands, such as those entered by a user.

17.   The applicant has argued that the disclosed objects pertaining to claim 19 are not defined during execution of a command processor through configuration commands loaded into the processor through a script or a command line, and that the disclosed objects are hard-coded, contrary to claim 19.

In response to this argument, please refer to the rejection of claim 19 *supra* and also to column 9, lines 39 – 42; and column 10, lines 36 – 44, wherein the defining of objects has been interpreted as sufficiently equivalent to building objects, and further wherein the commands pertaining to said defining, which assign functionality, may be interpreted configuration commands, such as those entered by a user, further wherein, as disclosed by column 6, lines 50 – 59, said interpreted configuration commands have been interpreted as sufficiently equivalent to

events interpreted that arise from the MOTIF toolkit that are reacted upon in a purely dynamic

manner, thus not hard-coded.

18.   The applicant has argued that Nahaboo et al. do not disclose <u>configuration commands</u>

that, <u>at run time</u>, build graphical objects for the graphical user interface and assign functionality

to the built graphical objects, that the objects are not built through such commands and are not

built at run time as defined in claim 24.

In response to this argument, please refer to the rejection of claim 24, *supra*, and also to

the response to the argument of section 12, *supra*.

19.   The applicant has argued that the elements of claim 24 would not be obvious to a person

of ordinary skill in the art since Nahaboo et al. disclose such a different method for altering an

interface.

In response to applicant's argument that Nahaboo et al. disclose such a different method

for altering an interface than Dangelo et al., it has been held that a prior art reference must either

be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular

problem with which the applicant was concerned, in order to be relied upon as a basis for

rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed.

Cir. 1992). In this case, both Nahaboo et al. and Dangelo et al. pertain to building objects in a

graphical user interface.

Furthermore, in response to this argument, a recitation of the intended use of the claimed

invention must result in a structural difference between the claimed invention and the prior art in

order to patentably distinguish the claimed invention from the prior art. If the prior art structure

is capable of performing the intended use, then it meets the claim.

Conclusion

20.   It is noted that any citation to specific, pages, columns, lines, or figures in the prior art

references and any interpretation of the references should not be considered to be limiting in any

way. A reference is relevant for all it contains and may be relied upon for all that it would have

reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332-

33,216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006,1009, 158

USPQ 275, 277 (CCPA 1968)).

21.   The prior art made of record and not relied upon is considered pertinent to the applicant's

disclosure. The cited documents represent the general state of the art.

22.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Eric A. Wiener whose telephone number is 571-270-1401. The

examiner can normally be reached on Monday through Thursday from 9am to 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Weilun Lo, can be reached on 571-272-4847. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications
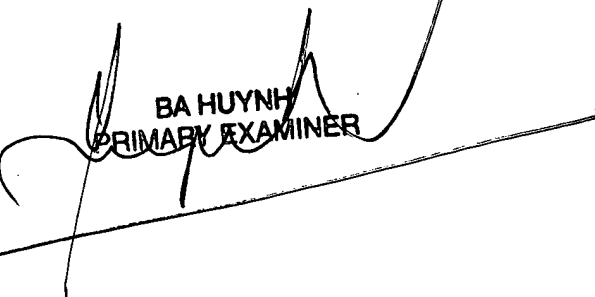
may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Eric Wiener
Patent Examiner
A.U. 2179

BA HUYNH
PRIMARY EXAMINER